

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
8 November 2001 (08.11.2001)

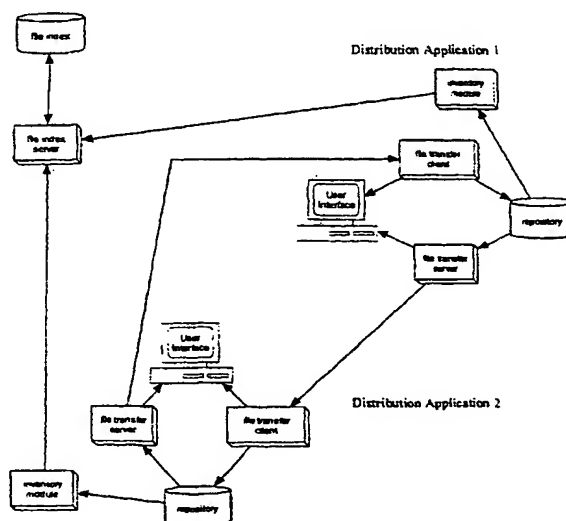
PCT

(10) International Publication Number
WO 01/84799 A2

- (51) International Patent Classification⁷: **H04L 29/00** (74) Agents: **HAVERSTOCK, Thomas, B.** et al.; Haverstock & Owens LLP, 260 Sheridan Avenue, Suite 420, Palo Alto, CA 94306 (US).
- (21) International Application Number: **PCT/US01/13271**
- (22) International Filing Date: **24 April 2001 (24.04.2001)** (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (25) Filing Language: **English**
- (26) Publication Language: **English**
- (30) Priority Data: **09/560,106** **28 April 2000 (28.04.2000)** **US** (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant: **NAPSTER, INC.** [US/US]; 1475 Veterans Blvd., Redwood City, CA 94063 (US).
- (72) Inventors: **FANNING, Shawn**; 33 Hayward Street, #104, San Mateo, CA 94402 (US). **FANNING, John**; 165 Nantasket Avenue, Hull, MA 02045 (US). **KESSLER, Edward**; 14505 Americh Hill Road, Los Gatos, CA 95032 (US).
- Published:**
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: **USE-SENSITIVE DISTRIBUTION OF DATA FILES BETWEEN USERS**



(57) Abstract: The invention contained herein describes a use-sensitive system for distribution of data files between users in a networked community of users. The system comprises each user having a distribution application that has at least one data file repository that has the ability to store at least one data file. The distribution application additionally includes a data file transfer server which makes available all data files located in the data file repository for download by other users, a file transfer client, and a user interface for displaying the actions and status of the distribution application to the user. In the system the file transfer client connects to the file transfer server and downloads a data file, and when the downloaded file is placed in the repository, the distribution application automatically makes available the downloaded data file to other distribution applications in the community of users.

WO 01/84799 A2



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Use-Sensitive Distribution of Data Files Between Users

By:

Shawn Fanning

John Fanning

Edward Kessler

Background

Many communities of Internet users share data files containing pictures, video, and audio data with one another. In most communities, there are typically just a few users who provide these interesting data files, and many more users who simply want to obtain them.

Users providing the files have typically done so either from altruism, from a desire to gain fame on the net in their particular net community, or from a desire to advertise a service and make money. Currently, various models for distribution of these types of files exist on the Internet.

One system for file distribution includes Usenet (Netnews), which allows a provider of a particular data file to post the data file to everyone that reads a particular newsgroup. This allows for widespread distribution of a posted data file, but all data files posted go to all recipients, and each recipient may select only from the files that were recently posted. All users have to look over all items posted for items they are interested in, and posted files are removed from the system after a short time, typically two or three days.

Another system for data file distribution includes a central website, allowing a data file owner to make his files available to all users on the network. While this allows recipients to select the files they wish to see, this approach does have a number of drawbacks. First, the community of users must be informed of the existence of the website. Secondly, each website is structured differently, depending on the whims of the provider of the data, making it difficult for requesting users to locate data they want. Third, and perhaps most importantly, since there are only a few website providers and many data file requesters, the few providers are deluged with requests, which overwhelms the provider's website. Regardless of the number of users that download a given file, the relative availability of that file is solely based on the single copy at the website. The more popular a file is, the less likely a given user will be able to obtain it because of the traffic generated by all the other users attempting to download it. This is true especially in regard to large data files such as video or audio data files.

An improvement on the central website model for distributing files involves data file mirroring. In this system, a central server contains a master copy of files to be shared, and each mirror server connects to the central server, and downloads all of the files to be mirrored. While this spreads the load amongst all the mirror servers, it is inherently centralized, containing an identical set consisting of both popular and unpopular files, and therefore is not sensitive to use. For instance, a file that 10,000 users have downloaded is just as widely available as a file that only 10 users have downloaded. Since mirroring is centrally administered and by definition is a complete copy of a central repository, it is not use-sensitive.

Yet another system for file distribution involves a chat room, where users of the chat room can exchange data that they are interested in trading in real time. This allows individuals to exchange data and does not require expensive, high-bandwidth central servers. However, because there are many requesters and few providers, the few
5 providers are deluged with requests. What is more, they must handle the requests individually and manually.

As a result of the problems listed above, few users in these communities wish to go through the effort and expense required in the previously discussed distribution models. Thus, the widespread availability of data files quickly turns into a one-for-one
10 exchange between specific parties.

Thus, it can be seen that there is a long-standing need on the Internet for a mechanism that facilitates the distribution of data files in a community of users.

The distribution system must provide users a way to quickly and reliably locate data they wish to acquire.

15 The distribution system must provide wide distribution for files that are popular, while at the same time provide limited distribution for files that are not of general interest.

Importantly, the mechanism for actually transferring the data file between users must be distributed, not centralized, so that bandwidth costs of providing data are fairly and properly distributed across the users of the community, and so that any user can
20 contribute new data files without relying on some central authority.

Summary

The invention contained herein describes a use-sensitive system for distribution of data files between users in a networked community of users. The system comprises each
25 user having a distribution application having at least one data file repository that has the ability to store at least one data file. The distribution application additionally includes a data file transfer server that makes available all data files located in the data file repository for download by other users, a file transfer client, and a user interface for displaying the actions and status of the distribution application to the user. In the system,
30 the file transfer client connects to the file transfer server and downloads a data file, and, when the downloaded file is placed in the repository, the distribution application automatically makes available the data file to other distribution applications in the networked community of users.

The system preferably has a file index server containing a file index, the file index
35 further having file descriptions of any data file in the at least one file repository of each distribution application. In an embodiment, the system further comprises an inventory module for automatically adding a file description to the file index when a new data file is added to the repository of any distribution application. The inventory module preferably automatically removes the file description in the file index when the file is removed from
40 the repository of any distribution application. The inventory module also preferably automatically removes all file descriptions from the file index associated with any distribution application when that distribution application is disconnected from the file index server.

In an embodiment of the invention, the system also includes an audio file player
45 module, wherein audio files located in the data file repository are audibly presented to the user.

In yet another embodiment, the distribution application has an audio file converter module, which constructs audio files and places them in the data file repository, thus making the new files immediately available to the community of users.

5 In an embodiment, the system further comprises a video file player module, wherein video files located in the data file repository are displayed visually to the user.

The system preferably has means for downloading a data file from a protected distribution application, wherein the requesting distribution application transmits a download request message to the index server, the index server forwards the download request message to the protected distribution application, the protected distribution
10 application initiates a connection to the requesting distribution application, and then transmits the file to the requesting distribution application.

The file description comprises any of the following: a title of the data file, the size of the data file, the type of data file, any text associated with the data file, the creator of the data file, the quality rating of the data file, and the distribution application where the
15 data file resides.

In a preferred embodiment, the system further comprises a search module for searching the file index, wherein a search request submitted by the distribution application is processed and results in a search response returned to that distribution application, wherein the search response comprises file descriptions matching the search
20 request. Preferably, the search module further comprises a filter module for pruning a search response of file descriptions that do not meet a limitation criteria. In an embodiment, the limitation criteria comprises a bandwidth limitation, wherein the search response is pruned of file descriptions from distribution applications that have a bandwidth capability that is below the bandwidth limitation. In an embodiment, the
25 search module comprises means for pruning the search response of file descriptions from distribution applications that will not accept further download requests. In a preferred embodiment, the search module comprises means for pruning the search response of file descriptions from the distribution application.

In an embodiment, the distribution application contains a sort module for sorting
30 the search response. Preferably, the search response is sorted using a responsiveness value, wherein the responsiveness value for another distribution application is determined by measuring the amount of time an echo reply message takes to be returned by the other distribution application to the user's distribution application. In an embodiment, the search response is pruned of file descriptions from any distribution application that did
35 not respond to the echo request within a predetermined period of time.

The data files are of the type selected from the group comprising: an audio data file, a text data file, an image data file, a video data file, a software executable data file, or a data file containing combinations, such as a multimedia data file.

In another embodiment, the invention further comprises an automatic selection
40 module, wherein a data file description is automatically selected and the associated data file is downloaded, the automatic selection module choosing a data file description from at least two candidate data file descriptions located on at least two file transfer servers using a scoring mechanism and then causing the file transfer client to initiate a download from the file transfer server with the best score. Preferably, the optimal score in a scoring
45 mechanism is calculated from any combination of the following factors: the file transfer server with a low round-trip response time, the file transfer server with a high network bandwidth, the file transfer server with a high percentage of successful downloads, and the file transfer server with the fewest active file transfers.

In yet another embodiment, the system further comprises a parallel download module, wherein different subsections of an identical data file are downloaded from at least two other distribution applications. Preferably, the at least two other distribution applications are automatically selected using a scoring mechanism.

5 The invention is a system of distribution applications that all incorporate both a file transfer client and a file transfer server. Each distribution application contains a repository of files to be provided to others on the network. In the preferred embodiment, any files downloaded to a distribution application are automatically added to that distribution application's repository of files, making newly downloaded files available for
10 re-export.

Joining the file transfer client and the file transfer server means that all users will act as suppliers of interesting data files to the community, including especially those users that are initially only interested in obtaining data files. As users are more successful at obtaining more files, they can in turn become providers of the files that they obtain to the
15 rest of the community.

The system of distribution applications is use-sensitive, because the community of users that interact using these distribution applications determines the particular files that flow through and exist on the network. Popular files that are downloaded frequently will exist in a large number of repositories. If a given file has had 10,000 users download it,
20 there are 10,000 places where other users can find this file, making it extremely unlikely that any one site will be overloaded with requests. Likewise, unpopular files will only exist in a couple of places, unlike a mirrored system that must store a copy of each file held by the central repository. However, if an unpopular file suddenly becomes popular, the system as a whole reacts rapidly to the change in usage to provide increased
25 availability for the file as demand increases, because each new user that successfully downloads the file becomes in turn a provider for the file.

Preferably, the invention also includes an index server containing the list of all the files on all the distribution applications. In this way, a user can obtain from the index server a high quality distribution application from which to download a desired file. All
30 distribution applications contribute a description of all files in their repositories to the index server when they first start up, and when a distribution application is disconnected, all of the descriptions associated with that distribution application are removed from the index server automatically, keeping the index server up to date.

35 **Brief Description of the Drawings**

Figure 1 shows an embodiment of the invention with two distribution applications communicating over a network.

Figure 2 shows a use sensitive embodiment of the invention with a number of distribution applications that distribute requested files between users.

40 Figure 3 shows an embodiment of the invention that uses a file index server to maintain file descriptions of files that are available for distribution.

Figure 4 is a preferred embodiment of a method of the invention.

Detailed Description

45 The invention as shown in Figure 1 comprises the following parts:

- A distribution application 10, comprising
 - a file repository 16,
 - a file transfer client 14,
 - a file transfer server 20,
 - a graphical user interface (GUI) 18.
- A network that interconnects the distribution applications.
- Optionally, in one embodiment, an index server that contains a list of file descriptions located on all distribution applications and a search module for searching the file descriptions.

As can be seen in Figure 1, all distribution applications 10 have the same functionality; one user's file transfer client 14 can download files from another user's file transfer server 20, and vice versa. So while at times only one of the directions will be explicitly specified, it is to be understood that anything that one user's distribution application can do, another user's distribution application 10 can do as well.

Data File Repository

The data file repository 16 or repository is the place where the user stores all of the data files to be shared with other users in the community. In the preferred embodiment, the data file repository is at least one directory on disk drives on a personal computer.

In one embodiment, the data file repository is a database; in another, it is a network accessible disk drive that the distribution application can access. In yet another embodiment, the repository is a collection of directories, enabling the user to organize files by type, by classification, or by attribute.

Files obtained from another distribution application 10 are initially stored in the data file repository 16 immediately after being downloaded, making these newly downloaded data files available to other distribution applications.

Inventory Module

A description of each file placed in the file repository 16 is automatically made available by an inventory module to other distribution applications 10 in the community. In the preferred embodiment, the inventory module verifies that each file is a valid file of the types of files available for distribution and extracts a title of the data file, the size of the data file, the type of data file, any text associated with the data file, the creator of the data file, and the quality rating of the data file. The inventory module is responsible for detecting the addition of a new file, making a description of it available to other distribution applications.

When a user does not wish to make a downloaded data file available for distribution, he may move the data file out of the file repository. The repository inventory module detects the removal of a file from the repository, and removes the associated file description. In one embodiment, the user is queried each time a file is downloaded, such as through a dialog box, whether or not this particular file is to be made available for distribution.

In one embodiment, the user can specify that all downloaded data files are not to be placed in the repository; in this case, newly downloaded data files will not automatically be shared with other distribution applications. The inventory module detects modifications to the repository in a manner that depends on the environment on which the distribution application 10 is executing.

In an embodiment, the inventory module uses operating system resources to automatically monitor the directory that contains the repository, receiving file add/file remove events from the operating system as files are added and removed.

In another embodiment, the inventory module must access the directory or directories periodically in order to detect modifications. In this embodiment, the inventory module periodically examines each directory comprising the repository, and when the last modification time shows that the directory has been modified recently, the inventory module checks the contents of the directory to ascertain which files if any have been added or removed.

In another embodiment, where the repository is a database, the inventory module uses database triggers to automatically monitor the repository, receiving file add and file remove events from the database as files are added and removed.

Other implementations of the inventory module for detecting addition and removal of files from a repository are possible, but fall within the definition of either employing the operating system to notify of file additions and removals, or periodically checking the repository to detect file additions and removals.

In an embodiment, the inventory module connects to an index server using TCP/IP (specified by RFC 798, herein included by reference). This index server contains a list of file descriptions on all the repositories of all the distribution applications currently in the community of users. When the connection to the index server is initially made, the inventory module transmits the description of each file in the repository to the index server. The index server adds each data file description reported by each inventory module to the master list of file descriptions.

As the inventory module detects file addition or file removal events in its repository, the inventory module transmits these events to the index server, which in turn creates or removes file descriptions from the master list of file descriptions as appropriate.

Upon disconnection of an inventory module from the index server, the index server removes all file descriptions that were previously transmitted to the index server by that inventory module. In this way, the index server is kept up-to-date as to the particular files that are accessible by distribution applications in the community. While a disconnection of a TCP/IP connection can occur for a variety of reasons, most often a disconnection is an indication that the distribution application 10 is unavailable to other users in the community. For example, a disconnection occurs when a distribution application is terminated or a network connection is terminated.

In one embodiment, periodic messages are sent by the index server using the TCP/IP connection to ensure that the inventory module is still alive. The frequency of these messages is once per minute, transmitted from the inventory module to the index server. If an index server does not receive a message within two minutes from a particular inventory module, it terminates the TCP/IP connection to that module, treating this event as a standard disconnection. This is necessary to detect situations where the machine on which the inventory module is executing crashes, or the network path to the inventory module goes down.

File Transfer Client

Each distribution application 10 has a file transfer client 14. The file transfer client allows the user of a distribution application to download files that are located in the repositories of other distribution applications. To initiate a transfer, the file transfer client
5 makes a connection to a file transfer server 20 using a stream-oriented connection protocol, such as TCP/IP.

Once a connection is made, the file transfer client transmits a download request to the file transfer server, the file transfer server determines if the requested file exists, and if it does, it transmits the file to the file transfer client 14. The progress of the transmission
10 is made available to each user interface of both distribution applications as sections of the file are received by the file transfer client. If the connection to the file transfer server is disconnected, both user interfaces reflect the change in connection status.

In one embodiment, in the case of a disconnection during transmission of a data file, where the data file is partially transmitted, the transfer may be re-initiated by the
15 downloading user using a retransmit request. In this case, the transfer resumes from the last data block received by the file transfer client 14. Therefore, a retransmit request is distinct from the standard download request, allowing the file transfer client to make use of the partial file previously downloaded from the file transfer server 20.

In another embodiment, a retransmit request is sent to a different file transfer
20 server that contains the identical file, in the case where the initial file transfer server is no longer available.

The specific messages sent between two applications engaged in performing some activity on a network is known as a protocol. To communicate with the file transfer server and to download files, the file transfer client 14 uses the FTP protocol specified in
25 RFC 959 by Postel & Reynolds, or another similar file transfer protocol. Basic file transfer protocols are well known in the art.

In one embodiment, a parallel download module improves transfer rates to slow file transfer servers 20 by connecting to at least two file transfer servers in order to download a given data file. The file to be downloaded from each file transfer server must
30 be identical. The parallel download module requests a different section of the given data file from each of the file transfer servers using a file subsection download request, and reports a completed download when all sections have been successfully downloaded. This allows a high-bandwidth file transfer client 14 to rapidly download parts of the same file from several lower bandwidth file transfer servers.

In another embodiment, the file transfer client transmits a reverse download
35 request to a protected file transfer server (a protected file transfer server is located behind a firewall and is unable to receive incoming connections from file transfer clients) by asking the index server to forward the reverse download request to the file transfer server using the file transfer server's inventory module index server connection. When a
40 protected file transfer server receives a reverse download request, it initiates a connection to the requesting file transfer client, after which the file download process occurs normally. This allows a file transfer client 14 to obtain a connection to a protected file transfer server.

File Transfer Server

The file transfer server 20 allows the user of the distribution application 10 to provide files that are located in its data file repository to other distribution applications.

The file transfer server handles incoming transmission requests from file transfer clients 14 of other distribution applications.

5 A file transfer client may request a file from a file transfer server. Upon receiving the request for a given file, the file transfer server examines its data file repository to determine if the file is available for downloading. If the file is available, it transmits that file to the file transfer client.

10 In a preferred embodiment, when a file transfer server 20 first starts executing, it makes an initial determination if it is protected by a firewall, or if it is otherwise unable to receive connections from file transfer clients. If so, it indicates this to the inventory module, which notifies the index server of its protected status. If this status is set, file transfer clients 14 must use reverse download requests that are forwarded through the index server to download files.

15 Each file transfer client that wishes to communicate with a file transfer server has its own TCP/IP connection to the file transfer server. In an embodiment, the specific protocol used by the file transfer server is a subset of the protocol embodied in the FTP protocol specification (RFC 959). As file transfer clients connect and request file downloads, the file transfer server 20 notifies the user of the status of each download using the user interface module. In a preferred embodiment, the distribution application limits the maximum number of concurrent download requests processed by its file transfer server. In addition, the distribution application 10 also specifies the maximum number of file transfer clients that may connect to its file transfer server. These limitations, and the current number of connections that are present and the number of downloads currently executing are transmitted to the file index server. When a new file transfer client 14 connects to a file transfer server, the limitations are checked, and connections in excess of this number are refused or queued.

25 In one embodiment, the file transfer server 20 handles a file subsection download request by downloading only that subsection of a given file, instead of providing the entire file. This allows a high-bandwidth file transfer client to download in parallel different parts of a given file from several lower bandwidth file transfer servers.

30 In another embodiment, the file transfer server 20 handles a reverse download request when it is protected, and a file transfer client 14 wishes to download a file. The reverse download request is issued by the client, transmitted to the index server, and forwarded down the distribution application's inventory module TCP/IP connection and is then given to the file transfer server. Upon receiving a reverse download request, the file transfer server actively connects to the file transfer client, and initiates the transfer of a particular file to the file transfer client. This "active connection" from file transfer server to file transfer client is the reverse of what normally occurs during a file download session.

40 Index Server

45 In one embodiment, an index server provides a master list containing data file descriptions of all data files that exist in repositories and available for download. The index server executes on a computer system that is remote from the distribution applications 10. The index server contains no data files itself; rather, it contains a list of data file descriptions located in repositories of currently executing distribution applications in the community.

Each distribution application has an inventory module (previously described) that

maintains a connection to the index server. When the inventory module initially connects to the index server, it uploads descriptions for each data file contained in its repository, as well as information regarding the configuration of its file transfer server 20.

5 As files are added and removed from the repository, the inventory module transmits a new description for each file added, and transmits a "delete" message for each file removed. If the connection between the index server and the file transfer server is broken, then the index server automatically removes the data file descriptions that the disconnected file transfer server had previously transmitted to the index server.

10 The index server optionally stores the following information for each file transfer server:

- IP address
- list of file descriptions
- number of current connections
- number of current downloads
- 15 connection limit
- download limit
- bandwidth to network
- protected by firewall
- percentage of successful downloads

20

The connection and download limits are placed by the user on his distribution application's file transfer server; they limit the number of concurrent connections and downloads that this file transfer server 20 will allow. In one embodiment, the network bandwidth configuration is set by the user, while in another embodiment the network bandwidth configuration is determined experimentally by the his file transfer server module. This information is transmitted by the file transfer server through the inventory module to the index server. The percentage of successful downloads, also known as the server reliability, is calculated by the index server. The index server does this by observing all transfers that occur from each file transfer server, and notes the number that succeed and the number that fail. From these, the index server calculates the percentage of successful downloads for each file transfer server.

30

The index server optionally stores information on each file description such as:

- filename
- file data type (text, audio, images, video, etc)
- 35 file size
- ancillary description (data type dependent)

40

The file data type can be text, audio, images, or video. Other types can be added without departing from the spirit of the invention. The file size is the number of bytes in the data file. For audio files, the ancillary description comprises the sampling rate, the bitrate, the creator of the file, and the original source from where the audio file was obtained. For text files, the ancillary description comprises the author, a synopsis, the

original publication date, and the ISBN. For video files, the ancillary description comprises the picture size, the running time, the bitrate, the number of frames per second, the synopsis, and the title of the video.

When a user wishes to locate a particular data file, the file index search module constructs a search request given the search criteria specified by the user. The search criteria can contain a complete filename or a subsection of the filename, limitations on any of the fields of ancillary data, the file size, or limitations on the file transfer server including bandwidth to network or percentage of successful downloads. The index server executes the search request, pruning the file descriptions as appropriate, and then the search response is displayed to the user.

In one embodiment, the file index search module automatically helps redirect traffic away from busy file transfer servers by sorting the file descriptions in the search results by the number of currently available downloads remaining for each of the file transfer servers. The number of currently available downloads remaining for a file transfer server is calculated by subtracting the number of current downloads from the download limit.

In an embodiment, the search index is comprised of several fault tolerant computer systems that act together to efficiently and rapidly process search requests and file description updates for a large number of users.

20

Graphical User Interface

The user interface 18 allows the user to configure the behavior of the file transfer client 14 and file transfer server 20, in addition to displaying the current status of all the components of the distribution application 10. The user can configure the maximum number of concurrent connections as well as the maximum number of concurrent downloads that the file transfer server will allow.

The user interface 18 displays the status of each file transfer as it occurs, along with the estimated time until completion, the filename, the percentage of the file transferred, and an identification string for each distribution application currently connected to the user's file transfer server. This information is displayed for both the file transfer server (listing other file transfer clients downloading files from the user's file transfer server) and the file transfer client (listing the user's file transfer client's downloads of files from other file transfer servers).

In one embodiment, the user interface provides a form for the user to construct a search request for a particular file type, based on the name of the file to be downloaded, ancillary data appropriate for that file type, and limitations on the file transfer server's bandwidth or percentage of successful downloads. The user interface 18 then transmits the search request to the index server, and receives a search response containing a list of file descriptions that matched the search request.

In another embodiment, the user interface 18 transmits an ICMP echo request to each file transfer server 20 that contains one or more of the returned file descriptions, and measures the amount of time it takes for each file transfer server to respond with an ICMP echo reply, yielding a file transfer server responsiveness value. If a file transfer server does not respond within a specified period of time, it is marked as not responding, and the file descriptions associated with that file transfer server 20 are pruned from the search response list.

Given a search response, the user can choose to sort the response by filename, server responsiveness, network bandwidth of the file transfer server, or by any one of the ancillary data fields appropriate to that particular data type.

5 Using the displayed sorted search response, the user can select one of the data file descriptions, thus initiating a download of the data file using the file transfer client 14. The user interface 18 shows the status of each download. Any download can be canceled prior to completion. Interrupted downloads are displayed as well.

Other Modules

10 In one embodiment, an automatic selection module utilizes the search response to allow the distribution application to automatically select the best file transfer server that contains the data file to be downloaded and initiates a download without requiring the user to select which particular file transfer server is to be accessed for the download. The automatic selection module determines the best file transfer server 20 based on a scoring
15 of each file description and the associated file transfer server description returned in the search response. The scoring calculation eliminates data file descriptions for files on file transfer servers that will not allow any more downloads, and weights the successful download percentage, the network bandwidth, and the echo response, applies this calculation to each file description, and returns the file transfer server with the optimal
20 score. The user's file transfer client is then instructed to perform the download from this file transfer server.

In another embodiment, a parallel download module facilitates the rapid download of files by simultaneously downloading different sections of the same file from at least two file transfer servers 20. The parallel download module selects the best of at least two
25 file transfer servers from among all the file transfer servers associated with the file descriptions in the search response. Then, the user's file transfer client is instructed to download different sections of the same file from each of the file transfer servers. This allows a user's distribution application with a high bandwidth connection to utilize several lower-bandwidth servers to maximize the transfer rate. It also distributes the file
30 transfer load across several different file transfer servers. Note that for this to function properly, both file transfer servers must have the identical file. In one embodiment file identity is assumed if the file descriptions (including the file size) are the same. Other methods of detecting file identity may be used here, and such methods are well known in the art.

35 In one embodiment, the distribution application 10 contains an audio file module, which includes an audio file player, as well as an audio file converter. The audio file player plays files located in the data file repository 16, while the audio file converter generates audio files either by sampling data from a microphone or tape, or by converting data stored on a CD-ROM or hard disk into a standard compressed audio file format.
40 Converted audio files are placed in the repository, allowing other users in the community to access these new audio files.

In another embodiment, the distribution application also contains a video file module, which includes a video file player, as well as a video file converter. Much as in the audio example above, video images (either still, or full motion) are converted from
45 external sources to compressed standard formats and are placed in the repository. Likewise, video files in the repository are displayed to the user by the video file player.

Claims

What is claimed is:

1. A use-sensitive system for distribution of data files between users in a networked community of users, the system comprising:
 - 5 a) each user having a distribution application further comprising:
 - i. at least one data file repository, with the ability to store at least one data file,
 - ii. a data file transfer server, which makes available all data files located in the data file repository for download by other users;
 - iii. a file transfer client; and
 - 10 iv. a user interface for displaying the actions and status of the distribution application to the user,
 - b) the file transfer client of a first user obtaining a connection to the file transfer server of a second user to download a data file,
wherein when a data file is placed in the data file repository of the first user, the
15 distribution application of the first user automatically makes available the data file to other distribution applications in the community of users.
2. The system of claim 1 further comprising a file index server containing a file index, the file index further having file descriptions of any data file in the at least one data file repository of each distribution application.
- 20 3. The system of claim 2, further comprising an inventory module for automatically adding a file description to the file index when a new data file is added to the repository of any distribution application.
4. The system of claim 3, wherein the inventory module automatically removes the file description in the file index when the file is removed from the repository of any
25 distribution application.
5. The system of claim 3, wherein the inventory module automatically removes all file descriptions from the file index associated with any distribution application when that distribution application is disconnected from the file index server.
6. The system of claim 1, further comprising an audio file player module, wherein audio
30 files located in the data file repository are audibly presented to the user.
7. The system of claim 1, the distribution application further comprising an audio file converter module, which constructs audio files and places them in the data file repository, thus making the new files immediately available to the community of users.
8. The system of claim 1, further comprising a video file player module, wherein video
35 files located in the data file repository are displayed visually to the user.
9. The system of claim 1, further comprising an image file display module, wherein image files located in the data file repository are displayed visually to the user.
10. The system of claim 1, further comprising a document file display module, wherein document files located in the data file repository are displayed to the user.
- 40 11. The system of claim 2, further comprising means for a requesting distribution application to download a data file from a protected distribution application, wherein the requesting distribution application transmits a download request message to the

index server, the index server forwards the download request message to the protected distribution application, the protected distribution application initiates a connection to the requesting distribution application, and then transmits the file to the requesting distribution application.

- 5 12. The system of claim 1, wherein the file description comprises any of the following: a title of the data file, the size of the data file, the type of data file, any alphanumeric data or images associated with the data file, the creator of the data file, the quality rating of the data file, and the distribution application where the data file resides.
- 10 13. The system of claim 2 further comprising a search module for searching the file index, wherein a search request submitted by the distribution application is processed and results in a search response returned to that distribution application, wherein the search response comprises file descriptions matching the search request.
14. The system of claim 13 further comprising a filter module for pruning a search response of file descriptions that do not meet a limitation criteria.
- 15 15. The system of claim 14 wherein the limitation criteria comprises a bandwidth limitation, wherein the search response is pruned of file descriptions from distribution applications that have a bandwidth capability that is below the bandwidth limitation.
- 20 16. The system of claim 13, further comprising a means for pruning the search response of file descriptions from distribution applications that will not accept further download requests.
17. The system of claim 13, further comprising a sort module for sorting the search response.
- 25 18. The system of claim 14, wherein the search response is sorted using a responsiveness value, wherein the responsiveness value for a second distribution application is determined by measuring the amount of time it takes for an echo reply message to make the round trip from the distribution application to the second distribution application and back again.
- 30 19. The system of claim 18, wherein the search response is pruned of file descriptions from any distribution application that did not respond to the echo request within a predetermined period of time.
20. The system of claim 1 wherein the data file is of the type selected from the group comprising: an audio data file, a text data file, a image data file, a video data file, and a software executable data file.
- 35 21. The system of claim 1 further comprising an automatic selection module, wherein a data file description is automatically selected and the associated data file is downloaded, the automatic selection module choosing a data file description from at least two candidate data file descriptions located on at least two file transfer servers using a scoring mechanism and then causing the file transfer client to initiate a download from the file transfer server with the best score.
- 40 22. The system of claim 21 wherein the optimal score in a scoring mechanism is calculated from any combination of the following factors; the file transfer server with a low round-trip response time, the file transfer server with a high network bandwidth and the file transfer server with a high percentage of successful downloads.
- 45 23. The system of claim 1 further comprising a parallel download module, wherein the distribution application downloads different subsections of an identical data file simultaneously from at least two other distribution applications.

24. The system of claim 23 wherein the at least two other distribution applications are automatically selected using a scoring mechanism.

25. A use-sensitive system for distribution of data files between users in a networked community of users, the system comprising:

- 5 a) each user having a distribution application, further comprising:
- i. at least one data file repository, having the ability to contain at least one data file,
 - ii. a data file transfer server, which makes available data files located in the data file repository for download by other users;
 - iii. a file transfer client; and
 - 10 iv. a user interface for displaying the actions and status of the distribution application to the user,
- b) the file transfer client of a first user obtaining a connection to the file transfer server of a second user to download a data file; and
- 15 c) a file index server containing a file index, the file index further having file descriptions of any data file in the at least one file repository of each distribution application,
- wherein when a data file is placed in the repository, the distribution application automatically makes available the data file to other distribution applications in the community of users.

20 26. The system of claim 25, further comprising an inventory module for automatically adding a file description to the file index when a new data file is added to the repository of any distribution application.

25 27. The system of claim 26, wherein the inventory module automatically removes the file description in the file index when the file is removed from the repository of any distribution application.

28. The system of claim 26, wherein the inventory module automatically removes all file descriptions from the file index associated with any distribution application when that distribution application is disconnected from the file index server.

30 29. A method for distributing data files between users in a networked community of users, each user having a distribution application, the method comprising:

- a) a first distribution application obtaining a connection to a second distribution application,
- b) the first distribution application requesting a data file from the second distribution application,
- 35 c) the second distribution application transmitting the data file to the first distribution application; and
- d) the first distribution application storing the data file into the first distribution application's data file repository,

40 wherein when the data file is placed in the first distribution application's data file repository, the first distribution application automatically makes available the data file to other distribution applications in the networked community of users.

30. The method of claim 29 further comprising each distribution application connecting to a file index, the file index having file descriptions of any data file in at least one data

file repository of each distribution application.

31. The method of claim 30, further comprising automatically adding a file description to the file index when a new data file is added to the repository of any distribution application.

5 32. The method of claim 31, further comprising automatically removing the file description from the file index when the file is removed from the repository of any distribution application.

33. The method of claim 30, further comprising automatically removing file descriptions associated with the distribution application from the file index when the distribution
10 application is disconnected from the file index.

34. The method of claim 29, further comprising audibly presenting to the user audio files located in the data file repository.

35. The method of claim 29, further comprising constructing audio files and placing them in the data file repository, making the new files immediately available to the
15 community of users.

36. The method of claim 29, further comprising visually displaying to the user video files located in the data file repository.

37. The method of claim 29, further comprising visually displaying to the user image files located in the data file repository.

20 38. The method of claim 29, further comprising displaying to the user document files located in the data file repository.

39. The method of claim 30, further comprising downloading a data file from a protected distribution application, wherein the requesting distribution application transmits a download request message to the index server, the index server forwards the download
25 request message to the protected distribution application, the protected distribution application initiates a connection to the requesting distribution application, and then transmits the file to the requesting distribution application.

40. The method of claim 30 further comprising searching the file index, wherein the distribution application submits a search request to the file index, the file index
30 performs the search, and responds to the distribution application with a search response comprising file descriptions matching the search request.

41. The method of claim 40 further comprising pruning the search response of file descriptions from distribution applications that have a bandwidth capability that is below a bandwidth limitation.

35 42. The method of claim 40, further comprising pruning the search response of file descriptions from distribution applications that will not accept further download requests.

43. The method of claim 40, wherein the search response is sorted using a responsiveness value, wherein the responsiveness value for a second distribution application is
40 determined by measuring the amount of time it takes for an echo message to make the round trip from a first distribution application to the second distribution application and back again.

44. The method of claim 43, wherein the search response is pruned of file descriptions from any distribution application that did not respond to the echo message within a
45 predetermined period of time.

- 5 45. The method of claim 40 further comprising automatically selecting the file description and downloading the associated data file, wherein the file description from at least two candidate data file descriptions located on at least two distribution applications are scored using a scoring mechanism, with the data file having the best score being downloaded.
- 10 46. The method of claim 45 wherein the optimal score in the scoring mechanism is calculated from any combination of the following factors; the distribution application with a low round-trip response time, the distribution application with a high network bandwidth and the distribution application with a high percentage of successful downloads.
47. The method of claim 29 wherein the distribution application downloads different subsections of an identical data file simultaneously from at least two other distribution applications, reducing the time required for the distribution application to receive the data file.

1/4

FIG. 1

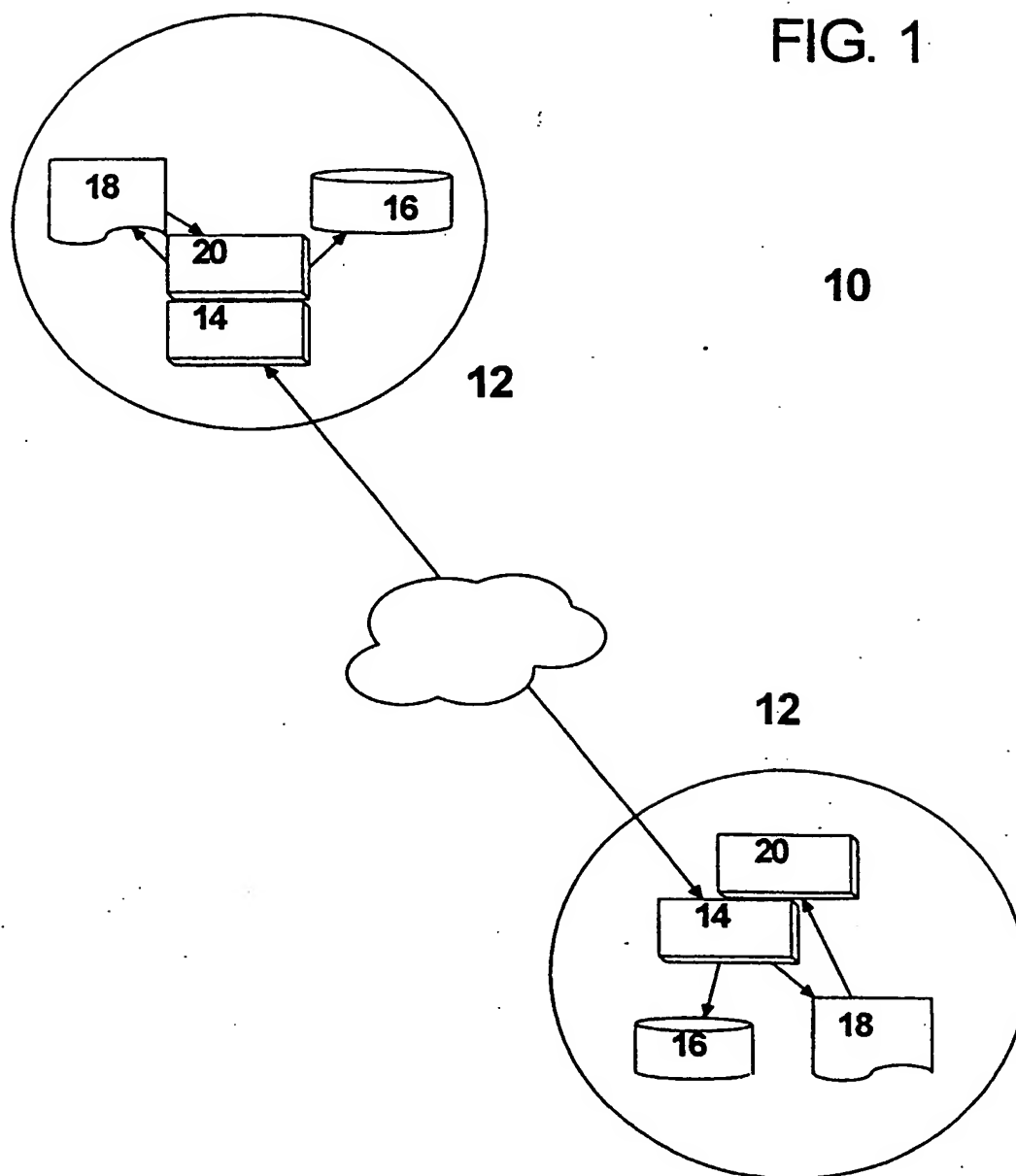


FIG. 2

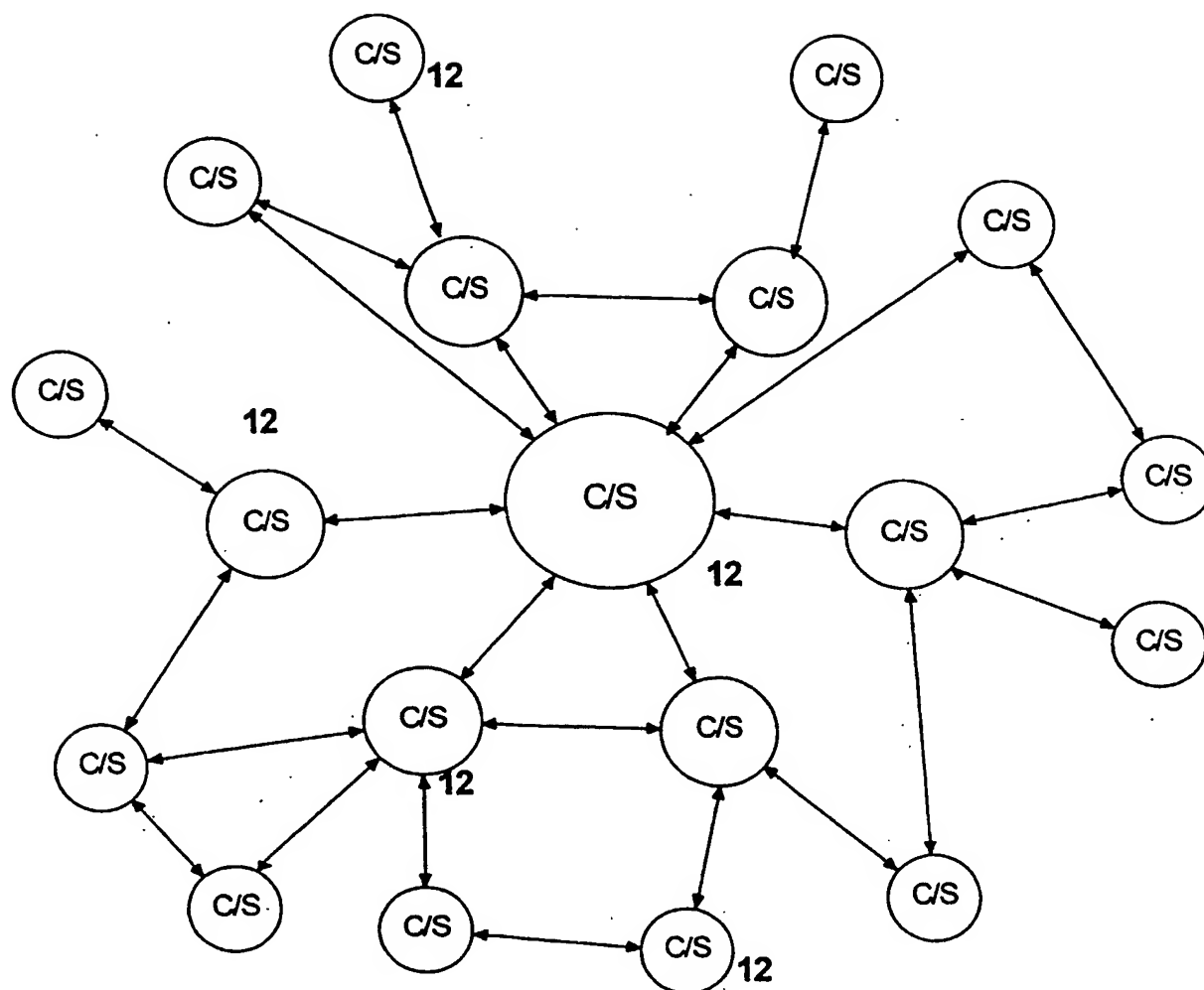
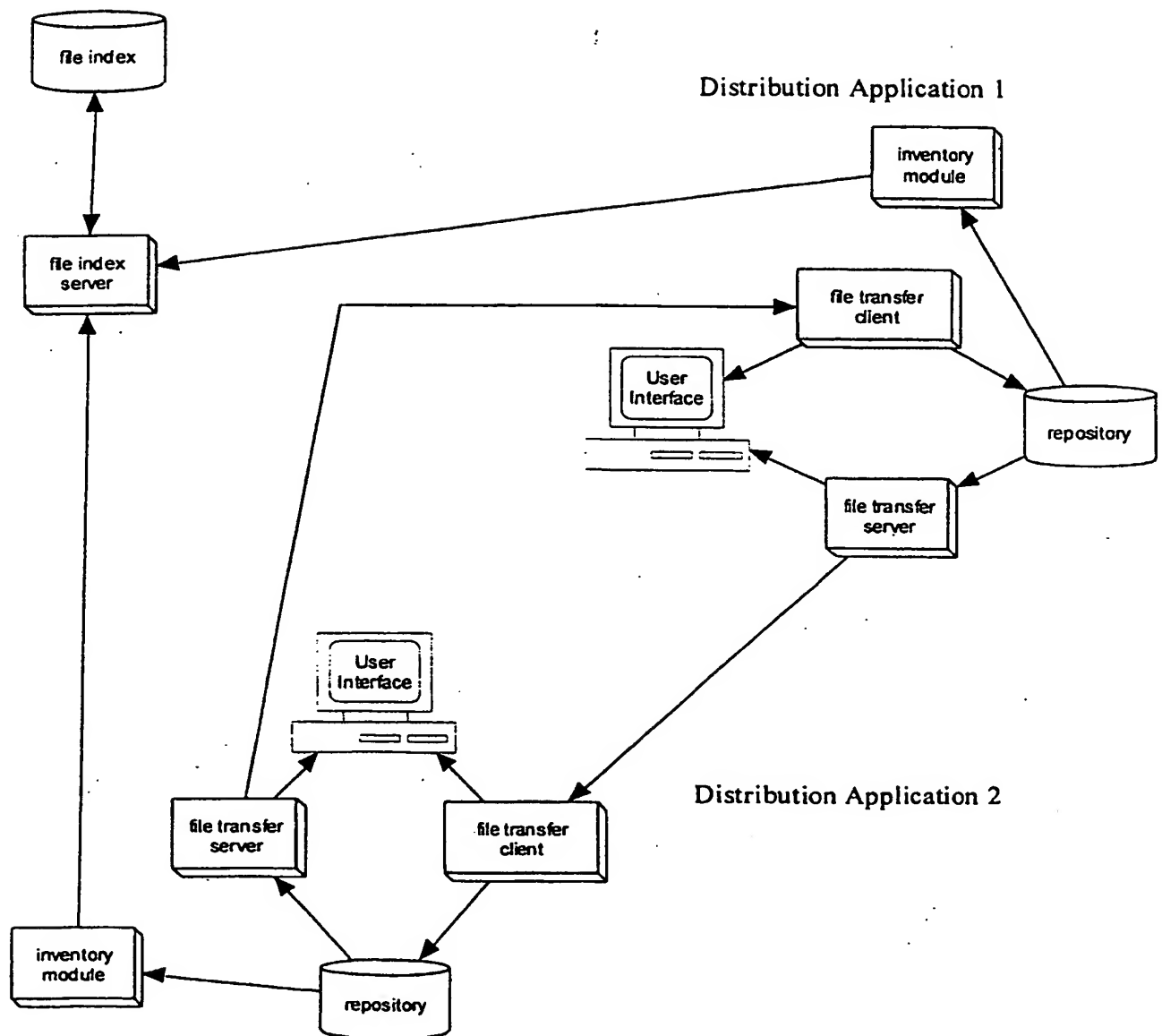


Fig. 3



4/4

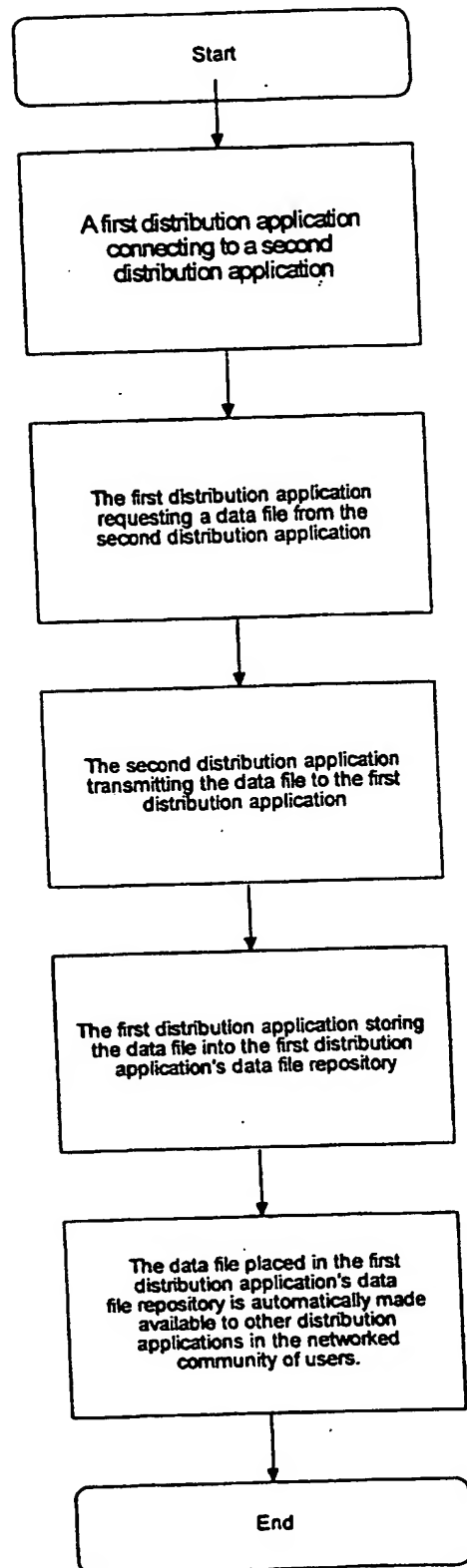


FIG. 4